REMARKS

Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 58-63, and 66-75 are pending in the application. Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 58-63, and 66-75 stand rejected. Claims 1, 11 and 33 have been amended. Claims 58-60 have been withdrawn and renumbered as claims 77- 79 to correct a typographical error.

Applicants request further review and examination in view of the claimed amendments.

Claim Rejections – 35 USC § 112

Claims 1-3, 6-15, 17-19

Claims 1-3, 6-15, 17-19 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite. The Examiner asserts that "the method is mixed with a programming syntax that is unrelated to the use of the computer, or the method for use." (Office Action, pg 2). Applicants respectfully disagree. The subject claims never used the phrase "programming syntax," but instead contain the recitation "programming language." A person skilled in the arts must know that programming languages must be related to the use of a computer and thus the claims are not indefinite. Therefore claim 1-3, 6-15, 17-19 are clearly directed to "a method for configuring a computer to generate computer executable instructions." This rejection should be withdrawn.

Claim 11

Claim 11 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite. Amendments have been made to correct any deficiencies related to the rejection. Reconsideration and withdrawal of the outstanding rejection under 35 U.S.C. § 112 is thus respectfully requested.

Claims 23-26, 28-37, 39-41

Claims 23-26, 28-37, 39-41 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite. The Examiner contends that the limitation is ambiguous because "there is no processor executing a high level language program. Such as Fortran program cannot be

executed by a processor. Interpret the modules are high level language programs." (Office Action, pg 3). Applicants disagree. The claims are clear on its face, since a person skill in the arts would know that the claims cannot possibly be interpreted as a high level language (such as Fortran) directly being executed on the processor. Even the Examiner himself concedes to the impossibility of this interpretation. Thus, the claims are not ambiguous. As for claim 33, appropriate Trademark label has been placed. Accordingly, these rejections should be withdrawn.

Claims 58-63, 66-75

Claims 58-63, 66-75 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite. In particular, the Examiner asserts that this method is unclear, whether it is for 'generating an object' by the preamble or describing an explicit interface member… The claims look like describing a high level language syntax than generation of an objection." (Office Action, pg 4). Applicants respectfully disagree. Claim 61, for example, clearly states that "*implementing, by the complier,... in response to* ***detecting the relationship*** *between the member and the name of the interface…*" Thus, steps are being performed "*in response to detecting the relationship between the member and the name of the interface.*" This is not a mere description of high level language syntax as proposed by the Examiner. Applicants submit that the claims 58-63, 66-75 are clear on its face. This rejection should be withdrawn.

Claim Rejections – 35 USC § 101

Claims 1-3, 6-15, 17-19

Claims 1-3, 6-15, and 17-19 stand rejected under 35 U.S.C. §101 because the Examiner asserts that the claims "as being a common method for using a computer to store a thing into a computer readable medium without show any practical application or a transformation of a subject matter….There is not practical application in the claims." (Office Action pg 5). Applicants respectfully submit that the Federal Circuit's *In re Bilski* decision has reaffirmed that the machine-or transformation test is the test for patentability of a process. *See, e.g., In re Bilski*, 545 F.3d 943, 20 (Fed. Cir. 2008) (en banc). In Bilski, the Court held that:

> "the proper inquiry under §101 is not whether the process claim recites sufficient "physical steps," but rather whether the claim meets the machine-or-transformation test… a claim that purportedly lacks any "physical steps" but is still tied to a machine or achieves an eligible transformation passes muster under §101." (*In re Bilski at 23.)* "The machine-or-transformation test is a two-branched inquiry; an applicant may show that a process claim satisfies §101 either by showing that his ***claim is tied to a particular machine***, or by showing that his claim transform an article." *(Id. at 10, citing Gottschalk v. Benson, 409 U.S. 63, 70, 93 S. Ct. 253, 34 L. Ed. 2d 273 (1972)).*(emphasis added).

As is clear from the claim language, all the operations as recited by claim 1 are tied to a machine. That is, the operations are performed by a computer. At least one computer device is required by claim 1 in order to "*implement an explicit interface member*" and thus is more than insignificant extra-solution activity. (*See Parker v. Flook, 437 U.S. 584, 590, 98 S. Ct. 2522, 57 L. Ed. 2d 451 (1978).* Accordingly, withdrawal of this rejection is respectfully requested.

Claim 23-26, 28-37, 39-41

Claims 23-26, 28-37, 39-41 stand rejected under 35 U.S.C. §101 because the Examiner argues that "under statutory medium, it requires the medium being the physical medium and tangible in a computer, and storing executable instruction so that when executable by a processor to do a practical thing." (Office Action, pg 5). Applicants respectfully disagree and request the Examiner to provide legal citation to support his arguments. As provided by the specification, a computer readable storage medium is clearly tangible and includes, for example, "RAM, ROM EEPROM, flash memory or other memory technology, CDROM, digital versatile disk (DVD) or other optical disk storage…" (*See* Specification, Pg 10, line 5-7). This rejection should be withdrawn.

Claim Rejections – 35 USC § 103

Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 58-63, and 66-75 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kramer *et. al.,* "Configuring Object-based Distributed Programs in REX, " Software Engineering Journal, 3-1992, pages 139-149, (hereafter "Kramer").

On page 7 of the Office Action, the Examiner explains his interpretation of Kramer:

> [Kramer] teaches using a mechanism (See p. 140, title) that is well defined with **interfaces (explicit interface member mechanism)** for configuring a computer to generate programming class (e.g. Fig. 4, p. 142). With this configuration language, in Fig. 4, it enables an object with **explicit interfaces (an explicit interface member)**. Using the configuration language, the object with explicit **interfaces** will not require declaring "public" to the interfaces.

Applicants respectfully disagree for three reasons:

First, while the Examiner equates interfaces with "explicit interface member mechanism" and "explicit interface member," the Examiner fails to point out where this is taught in the cited reference. The Examiner cites to "p140, title, " yet page 140 does not contain any title. Next, the Examiner cites to Figure 4. Similarly, Figure 4 provides no explanation in regards to how explicit interfaces are the same as "explicit interface member mechanism" and "explicit interface member." Finally, the two limitations are clearly two different terms, yet the Examiner merely treats them as the same term. Applicants respectfully request the Examiner to provide further citation and explanation from Kramer.

Second, Kramer's definition of "interfaces" suggests that an interface is only a set of methods and not an interface member. Kramer states: "Objects interact by method invocation. An object interface is usually described by those **methods** that it offers." (Pg 142, 2nd paragraph). In order words, each interface contains more than one method. (This definition is also consistent with what is well known in the arts.) Accordingly, "interfaces" in Kramer is not the same as the "explicit interface member" as recited in claim 1.

Finally, Kramer also fails to teach or suggest the phrase "*an implemented explicit interface **member** to be excluded from a public interface of said class"* as stated in claim 1.

The Examiner argues that "the object with explicit interfaces will not require declaring 'public' to the interfaces." (Office Action, pg 7). Applicants disagree. First, the Examiner once again fails to provide support for this assertion. Second, as discussed above, "interfaces" in Kramer is not the same as an "explicit interface member." Thus, excluding an "interface" from declaring public is not equivalent to "*an implemented explicit interface* ***member to be excluded*** *from a public interface of said class"* as recited in claim 1. Consequently, Applicants submit that Kramer also does not meet this further limitation.

Independent claims 23 and 61 contain limitations similar to those recited in claim 1. Consequently, Applicants submit that they also patentably define over Kramer for at least the reasons set forth above. Each of claims 2-3, 6-15, 17-19, 24-26, 28-37, 39-41, 62-63 and 66-79 depend directly or in directly from independent claims 1, 23 and 61, respectively, and are believed allowable for the same reasons.

CONCLUSION

Applicants request the Examiner reconsider the rejections and issue a Notice of Allowance of all the claims.

Date: October 13, 2009

/Amy O. Kwan/
Amy O. Kwan
Registration No. 59,829

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439